Remarks

Claims 1-13 remain pending. Claim 13 is currently amended. No new matter is being added.

Drawings

Corrected drawings for FIGS. 2A and 2B are hereby submitted. These corrected drawings replace "(Background Art)" with –Prior Art— as required by the Examiner in the latest office action. Therefore, applicants respectfully submit that the objection to the drawings is overcome.

Claim Objections

Applicants have hereby amended claim 13 in accordance with the Examiner's comments so as to overcome the objection to claim 13.

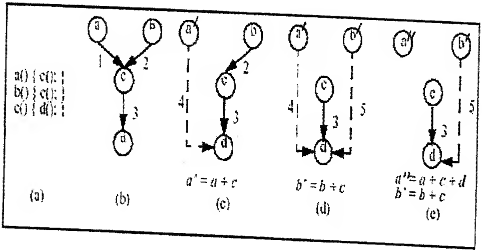
In regard to claim 7, applicants respectfully submit that previously-presented claim 7 recites, "memory configured to **store** computer-readable instructions and data" and "computer-readable instructions **stored** in said memory …." (Emphasis added.) Therefore, applicants respectfully submit that claim 7 already recites the "storing" language emphasized in the objection.

Claim Rejections--Section 103

Claims 1-13 were rejected under 35 U.S.C. 103 as being unpatentable over Kramskoy et al (US 7,080,366) (hereinafter "Kramskoy") in view of Berry et al (US 6,651,243) (hereinafter "Berry"). This rejection is respectfully traversed.

Claim 1 relates to **inline specialization**. In particular, as recited in claim 1, "if multiple call-chains in the **call-graph** have a **common call site**, inlining the common call site in one or more of the **call-chains**, without inlining the common call site into all of said multiple call-chains having the common call site." (Emphasis added.)

The claimed inline specialization is illustrated in Figure 6 which is reproduced below for convenience of reference.



```
a() { c(); }
b() { c(); }
c() { d(); }
```

Figure 6

The above figure is discussed starting on page 26, line 25, as follows. "We now discuss the creation of new edges and the consequences thereof in further detail. To visualize the call-relationship after a routine is inlined into another, we refer to Figure 6. Figure 6(a) shows a simple code segment. Figure 6(b) shows the corresponding call-graph. Figure 6(c) shows the call graph after a decision is taken to inline the call site *1*. A new edge *4*, shown as a dotted edge, is added from node *a* (now *a'*) to node *d*. This is because when node *a* inlines node *c*, the call to node *d* gets imported into *a*. This new call site, *4,* is inserted into the work-list and is considered by subsequent inline analysis. Figure 6(d) shows the call-graph after a decision is taken to inline the call site 2. A new edge 5, shown as a dotted edge, is added from node *b* (now *b'*) to node *d*. It is to be noted that subsequent to the scenario shown in Figure 6(d), the inline analysis phase could decide to inline only one of call sites *4* and *5*. **If call site *4* is inlined without inlining call site *5*, this would achieve inline specialization, as shown in Figure 6(e). This is because node *a* (now *a''*) will have inlined both nodes *c* and *d*, while node *b* (denoted *b'*) will have inlined node *c* alone.**" (Emphasis added.)

In contrast, while Kramskoy relates generally to a compiler, the cited portions of Kramskoy per the latest office action have *nothing* to do with inlining call sites in a call graph, much less inline specialization. The following discusses some of the stark deficiencies in the disclosure of Kramskoy in relation to claim 1.

First, FIG. A5 and portions of the text are cited against the limitation of being "given a call-graph". However, FIG. A5 and the cited text refer to a **call stack**, not a **call graph**. Both a "call stack" and a "call graph" are terms of art in the computer programming field. They mean completely different things.

The claimed **call graph** is a graph that **indicates relationships between routines** of a program. Examples of call graphs are shown, for example, in FIG. 2B of the present application and are discussed in the related text as follows. "The nodes of the call graph correspond to the program routines, and the edges of the call graph correspond to the call sites. The call site labeled '1' corresponds to the call from main() to foo(). The call site labeled '2' corresponds to the call from main() to bar()....." (Page 7, lines 22-25.)

In contrast, a **call stack** is a data structure which **stores information about actively executing routines** of a program. A call stack may also be referred to as a run-time stack or an execution stack.

Second, three **code blocks** 1072, 1080, 1084 in FIG. 1E are cited against the claimed "common call site". For example, code block 1072 precedes a conditional branch, and code block 1080 precedes a call/invoke. However, these code blocks are **not common call sites**. A common call site is one such as call site "3" in the above-discussed Figure 6 of the present application.

Third, **dominant path** 1088 in FIG. 1E is cited against the claimed "call-chain". However, the dominant path 1088 is **not a call-chain** (i.e. a chain or series of calls). Instead, the dominant path 1088 appears to not have any calls within it. For example, the dominant path 1088 does not follow the call along C to code blocks 1082.

For at least the above-discussed reasons, applicants respectfully submit that the cited portions of Kramskoy does not disclose or suggest the above-discussed limitations of claim 1.

In regard to Berry, Berry is cited in relation to FIGS. 11 and 12 as disclosing a call graph. However, applicants respectfully submit that FIGS. 11 and 12 of Berry show **call stacks, not call graphs**. Call stacks, whether in tree or table form, are not call graphs. As discussed above, a call graph indicates calling relationships between routines of a program, while a call stack is a data structure which stores information about actively executing routines of a program. Thus, Berry does not cure the many deficiencies of Kramskoy which are discussed above.

Thus, applicants respectfully submit that claim 1 is patentably distinguished over Kramskoy in view of Berry and so overcomes this rejection.

Claims 2-6 depend from claim 1. Hence, claims 2-6 also overcome this rejection for at least the reasons discussed above in relation to claim 1.

Independent claim 7 recites similar limitations to those discussed above in relation to claim 1. In particular, claim 7 recites "computer-readable instructions stored in said memory and configured to **inline a common call site** in one or more **call-chains** in a **call-graph**, without inlining the common call site into all call-chains having the common call site." (Emphasis added.) Hence, for at least the above-discussed reasons, claim 7 also clearly overcomes this rejection.

Claims 8-12 depend from claim 7. Hence, for at least the above-discussed reasons, claims 8-12 also clearly overcome this rejection.

Independent claim 13 also recites similar limitations to those discussed above in relation to claim 1. In particular, claim 13 recites "an inline specialization feature such that given a call-graph, if multiple **call-chains** in the **call-graph** have a common call site, the **common call site is inlined** in one or more of the call-chains, without having to inline the common call site into all of the multiple call-chains having the common call site." (Emphasis added.) Hence, for at least the above-discussed reasons, claim 13 also clearly overcomes this rejection.

Conclusion

For the above-discussed reasons, applicant respectfully submits that the objections and rejections of the office action are overcome by the claims as they now stand amended. Favorable action is respectfully requested.

The Examiner is also invited to call the below-referenced attorney to discuss this case.

Respectfully Submitted,

Dhruva Ranjan Chakrabarti, et al.

Dated: October 13, 2008

/James K. Okamoto, Reg. No. 40,110/
James K. Okamoto, Reg. No. 40,110
Okamoto & Benedicto LLP
P.O.Box 641330
San Jose, CA 95164-1330
Tel: (408) 436-2111
Fax: (408) 436-2114

CERTIFICATE OF MAILING

I hereby certify that this correspondence, including the enclosures identified herein, is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below. If the Express Mail Mailing Number is filled in below, then this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service pursuant to 37 CFR 1.10.

Signature:			
Typed or Printed Name	James K. Okamoto	Dated	10/13/2008
Express Mail Mailing Number (*optional*):			